The HPEC Challenge Benchmark Suite

Ryan Haney, Jeremy Kepner, and Theresa Meuse

Massachusetts Institute of Technology
Lincoln Laboratory

HPEC 2006

This work is sponsored by the Defense Advanced Research Projects Agency under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.
## Motivation

### Advanced Sensor Platforms

- Single-processor operations
- Taken from PCA program

### Processor and System Architectures

- **Single Processor Element**
- **Tiled Processors**
- **Multi-computers**

### System Analysis and Design

**Implement Benchmarks**
- Design
- Code
- Tune

**Measure Performance**
- Throughput
- Power
- Stability

**Design System**
- Choose components
- Hardware size
- Required software performance

### HPEC Challenge Benchmark Suite

**Kernel Benchmarks**
- Single-processor operations
- Taken from PCA program

**SAR Application Benchmark**
- Multi-processor compact application
- Taken from HPCS program
Outline

• Motivation

• Benchmark Suite
  – Kernel Benchmarks
  – SAR Application Benchmark

• Web Site
## Signal and Image Processing Kernels

### FIR
- **Input Matrix**: M Channels
  - Data Set 1: M Filters (~10 coefficients)
  - Data Set 2: M Filters (>100 coefficients)
- **Bank of filters applied to input data**
- **FIR filters implemented in time and frequency domain**

### SVD
- **Input Matrix**: M Channels
  - **Bidiagonal Matrix**
  - **Diagonal Matrix Σ**
- **Produces decomposition of an input matrix, X=UΣV^H**
- **Classic Golub-Kahan SVD implementation**

### QR
- **A (MxN)**
  - **Q (MxM)**
  - **R (MxN)**
- **Computes the factorization of an input matrix, A=QR**
- **Implementation uses Fast Givens algorithm**

### CFAR
- **Dopplers**
  - **C(i,j,k)**
- **Range**
  - **T(i,j,k)**
- **Target List**
  - (i,j,k)
- **Creates a target list given a data cube**
- **Calculates normalized power for each cell, thresholds for target detection**

---

*MIT Lincoln Laboratory*
**Information and Knowledge Processing Kernels**

### Genetic Algorithm
- Selection
- Crossover
- Mutation
- Evaluate each chromosome
- Select chromosomes for next generation
- Crossover: randomly pair up chromosomes and exchange portions
- Mutation: randomly change each chromosome

### Pattern Match
- Compute best match for a pattern out of set of candidate patterns
  - Uses weighted mean-square error

### Database Operations
- Three generic database operations:
  - search: find all items in a given range
  - insert: add items to the database
  - delete: remove item from the database

### Corner Turn
- Memory rearrangement of matrix contents
  - Switch from row to column major layout
SAR Benchmark Computational Challenges and Data Products

Front-End Sensor Processing
Scalable Data and Template Generator
- Raw SAR
- Templates

Kernel #1 Image Formation
- SAR Image

Template Insertion
- SAR Image
- Templates

Back-End Knowledge Formation
Kernel #4 Detection
- Detections

Validation

Scalable synthetic data generation
- Pulse compression
- Polar Interpolation
- FFT, IFFT (corner turn)

Sequential store
- Non-sequential retrieve
- Large & small IO

Large Images difference & Threshold
- Many small correlations on random pieces of large image
Outline

• Motivation
• Benchmark Suite
  – Kernel Benchmarks
  – SAR Application Benchmark
• Web Site
HPEC Challenge Web Site: 
http://www.ll.mit.edu/HPECchallenge

• Purpose:
  – Allow exploration of performance results for a wide range of architectures

• Provide public and password protected pages
  – Secure potentially sensitive architecture results

• Public pages:
  – Benchmark information
  – Benchmark software
  – Periodic releases of benchmark results
Benchmark Information

Benchmarks

The HPEC Challenge benchmark suite consists of a set of eight kernel benchmarks and a System Analysis (SAR) system benchmark for quantitatively comparing HPEC systems. The kernel benchmarks are designed to reflect a broad range of DoD signal and image processing applications. The scalable SAR is one of the most common functions in DoD surveillance systems. In addition, it includes standard class applications.

Metrics of interest for the benchmarks are defined on the metrics page.

Kernel Benchmarks

1. Time-Domain Finite Impulse Response Filter Bank - TDFIR
2. Frequency-Domain Finite Impulse Response Filter Bank - FDFIR
3. QR Factorization - QR
4. Singular Value Decomposition - SVD
5. Constant False-Alarm Rate Detection - CFAR
6. Pattern Matching - PM
7. Genetic Algorithm - GA
8. Database Operations - DB
9. Corner Turn - CT

Multi-Processor Application Benchmark

1. Synthetic Aperture Radar - SAR
Software

Download: HPEC Challenge Benchmark Suite v1.0 (Kernel Benchmarks only)
Download: HPEC Challenge SAR Benchmark v1.0 (Found under SSAC #3 v1.0)

Contents:

Version 1.0 of the benchmark suite download contains the kernel-level benchmarks only. A future release will also include a SAR multi-processor application benchmark. Information on the kernel-level benchmarks may be found on this page. For now, the SAR benchmark must be downloaded from the HPCS program web page. This will require the HPCS program.

Software Instructions

Kernel Benchmarks

The following descriptions and instructions can be found in the README.txt file found in the hpecChallenge/kernelBenchmark directory after unzipping the software distribution file. To unzip the distribution file, use winzip in Windows, or "gunzip file.tar.gz; tar xvf file.tar" in Unix.
• Registration
  – Obtain username, access to password protected pages

• Password protected pages:
  – Upload results to site
  – Architecture results, reports
Welcome Ryan Haney

This page allows a user to upload benchmark results to the HPEC Challenge website to be shared with other users. To upload results to the website, you must first enter information on the system that was benchmarked. The following sections allow you to add a new system along with relevant benchmark results to the site, or edit an existing system that you have added in the past.

Create a New System

The following link will prompt you for general information concerning the benchmarked system. A subsequent page will allow you to add more specific configuration details, and the actual benchmark results. No information will be viewable by other users until submitted to the site. This option will be provided on a subsequent page. System and result information may be deleted and removed from the site at any time.

- Create New System

Edit an Existing System

The following links allow you to add or edit system configuration details for systems already added to the site.

- Mercury PowerPC G4 System - 550 MHz G4 processor. Future documentation...
- Intel Xeon System - 2.8GHz Xeon processor...

Step 1

- Create a new system
- i.e. Describe the system you benchmarked
Step 2

- Fill in more detailed system specifications

What Now?

1. Enter in the information about the system’s configuration.
2. Upload or manually enter the results the system achieved on the HPEC Challenge benchmark suite.
3. Submit for review. The system information and results will not be publicly viewable until it has been submitted.
Uploading Results

Step 3

- Upload results from benchmark results file, or manually fill in

---

**System Overview**

<table>
<thead>
<tr>
<th>User</th>
<th>Ryan Haney</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>MIT Lincoln Laboratory</td>
</tr>
<tr>
<td>Interface</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Power (W)</td>
<td>50</td>
</tr>
</tbody>
</table>

**What Now?**

1. Enter in the information about the system's configuration.
2. Upload or manually enter the results the system achieved on the HPEC Challenge benchmark suite.
3. Submit for review. The system information and results will not be publicly viewable until it has been submitted.

**Optimized Code Results**

**Reference Code Results**

**Benchmark Results: Manual Entry**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Dataset</th>
<th>Reference Code Latency (s)</th>
<th>Optimized Code Latency (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPAR</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>MDPER</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>OR</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CFA</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>IM</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>GA</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>OT</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>DB</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
http://www.ll.mit.edu/HPECchallenge

Viewing Results

- General comparisons between architectures
- Detailed results and specifications for architectures

### Optimized Code Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Dataset</th>
<th>Workload (Miop/s)</th>
<th>Latency (s)</th>
<th>Throughput (Miop/s)</th>
<th>Throughput/Watt (Miop/s/W)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Domain FIR Filter Bank</td>
<td>2</td>
<td>1.27</td>
<td>0.0019</td>
<td>1227.0</td>
<td>207.4</td>
<td>1.000</td>
</tr>
<tr>
<td>Frequency Domain FIR Filter Bank</td>
<td>1</td>
<td>33.55</td>
<td>0.0431</td>
<td>729.0</td>
<td>155.8</td>
<td>1.000</td>
</tr>
<tr>
<td>QR Factorization</td>
<td>1</td>
<td>3.83</td>
<td>0.6910</td>
<td>781.0</td>
<td>123.4</td>
<td>0.956</td>
</tr>
<tr>
<td>QR Factorization</td>
<td>2</td>
<td>30.55</td>
<td>0.5005</td>
<td>504.0</td>
<td>126.8</td>
<td>0.956</td>
</tr>
<tr>
<td>QR Factorization</td>
<td>3</td>
<td>45.00</td>
<td>0.3763</td>
<td>595.0</td>
<td>116.0</td>
<td>0.956</td>
</tr>
<tr>
<td>Singular Value Decomposition</td>
<td>2</td>
<td>20.27</td>
<td>0.0500</td>
<td>403.0</td>
<td>61.0</td>
<td>0.064</td>
</tr>
<tr>
<td>Singular Value Decomposition</td>
<td>3</td>
<td>153.87</td>
<td>0.7181</td>
<td>214.0</td>
<td>42.8</td>
<td>0.064</td>
</tr>
<tr>
<td>Constant False-Alarm Rate</td>
<td>1</td>
<td>0.17</td>
<td>0.0011</td>
<td>115.0</td>
<td>31.2</td>
<td>0.004</td>
</tr>
<tr>
<td>Constant False-Alarm Rate</td>
<td>2</td>
<td>0.50</td>
<td>0.0025</td>
<td>203.5</td>
<td>40.7</td>
<td>0.004</td>
</tr>
<tr>
<td>Constant False-Alarm Rate</td>
<td>3</td>
<td>0.02</td>
<td>0.0002</td>
<td>94.5</td>
<td>19.3</td>
<td>0.004</td>
</tr>
<tr>
<td>Pattern Matching</td>
<td>1</td>
<td>1.20</td>
<td>0.0595</td>
<td>115.0</td>
<td>22.0</td>
<td>0.094</td>
</tr>
<tr>
<td>Pattern Matching</td>
<td>2</td>
<td>1.10</td>
<td>0.1772</td>
<td>116.0</td>
<td>23.2</td>
<td>0.094</td>
</tr>
<tr>
<td>Genetic Algorithm (GA)</td>
<td>1</td>
<td>0.01</td>
<td>0.0001</td>
<td>149.9</td>
<td>30.0</td>
<td>0.311</td>
</tr>
<tr>
<td>Genetic Algorithm (GA)</td>
<td>2</td>
<td>0.50</td>
<td>0.0025</td>
<td>201.5</td>
<td>40.7</td>
<td>0.311</td>
</tr>
<tr>
<td>Genetic Algorithm (GA)</td>
<td>3</td>
<td>0.02</td>
<td>0.0002</td>
<td>94.5</td>
<td>19.3</td>
<td>0.311</td>
</tr>
<tr>
<td>Genetic Algorithm (GA)</td>
<td>4</td>
<td>0.11</td>
<td>0.0010</td>
<td>61.0</td>
<td>12.7</td>
<td>0.311</td>
</tr>
</tbody>
</table>
Summary

• The HPEC Challenge is a publicly available suite of benchmarks for the embedded space
  – Representative of a wide variety of DoD applications

• Benchmarks stress computation, communication and I/O

• Benchmarks are provided at multiple levels
  – Kernel: small enough to easily understand and optimize
  – Compact application: representative of real workloads
  – Single-processor and multi-processor

• See http://www.ll.mit.edu/HPECchallenge/ for benchmark documentation and software

• Please send feedback to hpecChallenge@ll.mit.edu